Osciladores, expr, if

En esta lección vamos a ver varios tipos de **ondas**, que, por ejemplo, se utilizan para generar sonido en sintetizadores. Tambien vamos a ver el objeto "**expr**" y el condicional "**if**". Encontrareis todos los patches de demostración en la carpeta de material correspondiente a esta lección.

Onda sinusoidal. osc~

https://giphy.com/embed/pS8uMLP7sgXmK8d3Mr/video

Figura 1. Persona oscilando y generando un movimiento periódico.

El objeto "**osc~**" es un generador de ondas sinusoidales, también conocidas como ondas puras.

Hemos visto este objeto en la Practica 2: nuestro pr... | Librería CATEDU, pero vamos a recordarlo un poco. Debe recibir en su **inlet izquierdo** la **frecuencia** de la onda que queramos generar en Hz. Este objeto va a generar una onda pura, esto quiere decir una onda sin armónicos, constituida por una sola frecuencia y con una **amplitud** de **1**, por lo que generara valores **de -1 a 1**.

Copyright 2026

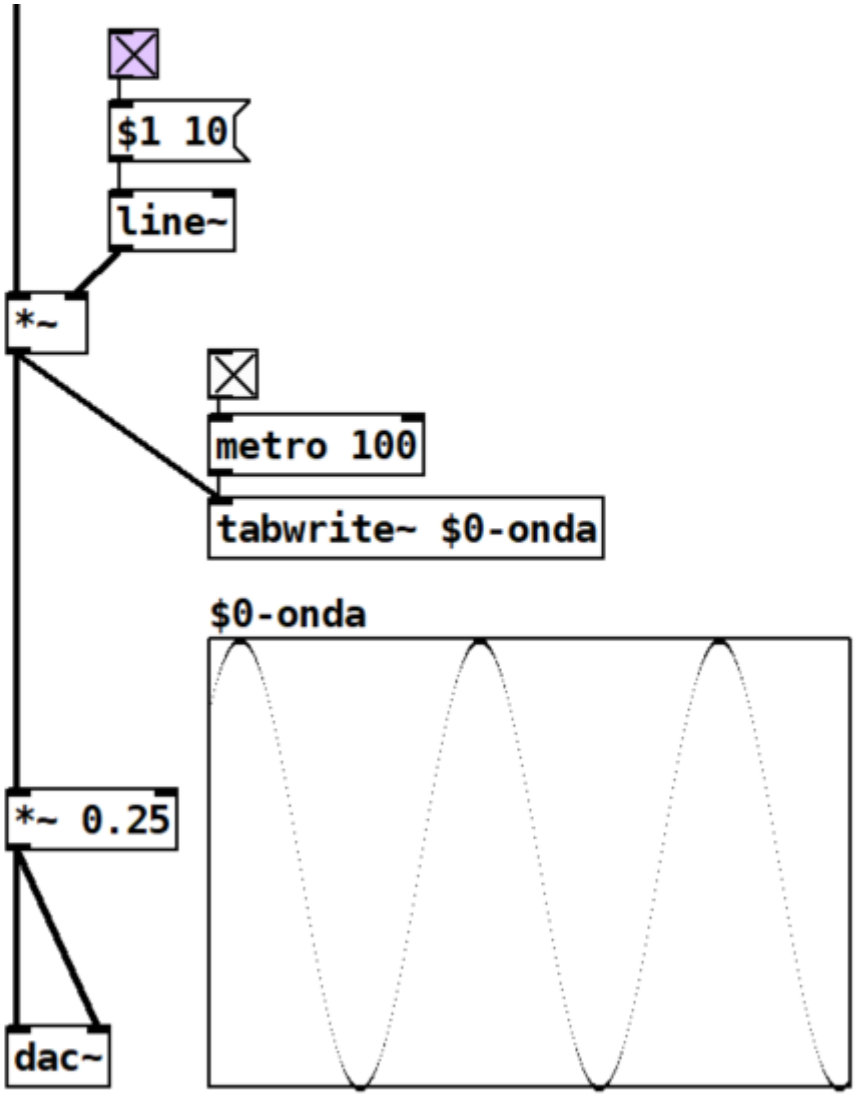


Figura 2.

patch *Onda-Sinusoidal.pd*

Onda de sierra: phasor

Estas ondas suenan más agresivas y estridentes que las ondas sinusoidales. La forma de esta onda son una **sucesión de rampas** iguales que van **de 0 a 1** en un tiempo determinado. Estas rampas

están generadas por el objeto "**phasor~**". Al llegar a 1 los valores caen inmediatamente a 0 desde donde comienza de nuevo la rampa. Cada rampa será una repetición y la frecuencia de esta onda vendrá determinada por el número de rampas sucedido en una unidad de tiempo. Tenemos una onda de sierra que va de 0 a 1, luego su **amplitud** sera de **0.5**, por lo que vamos a **normalizarla**, que quiere decir ajustar el centro del desplazamiento transversal de una partícula en 0 con una amplitud de 1. Para ello la multiplicaremos por dos para obtener una amplitud de 1 y para colocar su centro en 0 le restaremos 1.

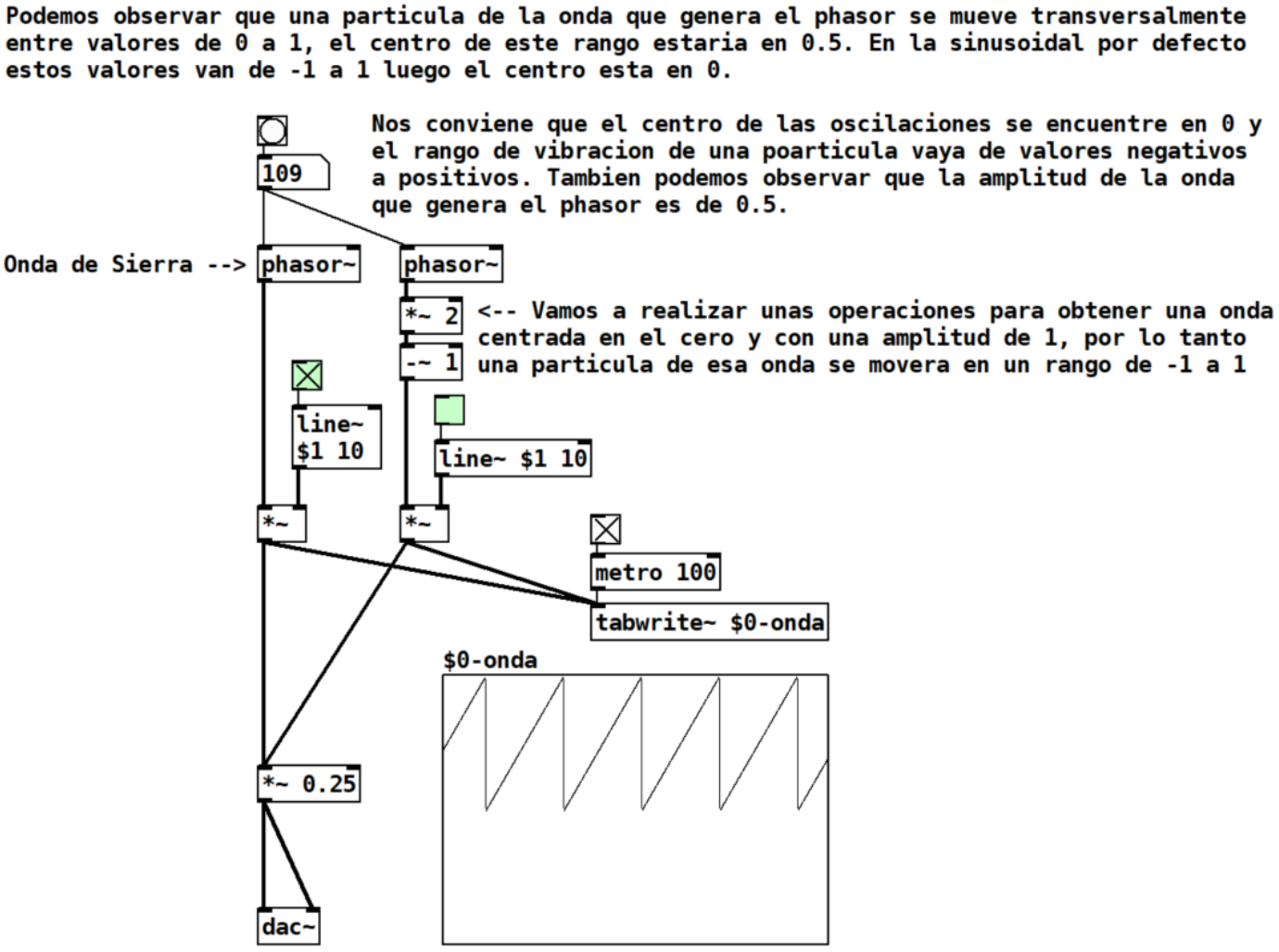


Figura 3. patch *Onda-Sierra.pd*

Los **toggles verdes** activan y desactivan la **onda sin normalizar** y la **onda normalizada** para que podáis compararlas en el patch *Onda-Sierra.pd*. Si encendéis los dos a la vez tendréis una combinación de ambas ondas.

Copyright 2026

Onda rectangular: phasor + filtro

Su forma se tiene que alternar entre -1 y 1, generando una **onda rectangular**. Para conseguir esta onda vamos a partir de una onda de sierra que vamos a transformar con el objeto "**expr**". Abre el patch Expr-explicacion.pd y vamos a ver un poco cómo funciona el objeto expr:

expr

Este objeto nos va a permitir **evaluar expresiones** o realizar operaciones utilizando operadores matemáticos (**<**, **==**, **+**, ...) o **estructuras condicionales** (si esta condición se cumple, entonces envía en valor A, si esta condición no se cumple envía el valor B. En programación por defecto las situaciones ciertas se representan con un 1, y las falsas con un 0.

En las expresiones cuando utilicemos el **$** vamos a especificar el **tipo de variable** que es (int, float o symbol). Cuando sea un int colocaremos una **i entre el símbolo $ y el número** que representas la posición ocupa la variable en la expresión. ej $i2 Cuando sea un float colocaremos un f y para symbol una s.

$Tipo-de-variablePosicion-variable

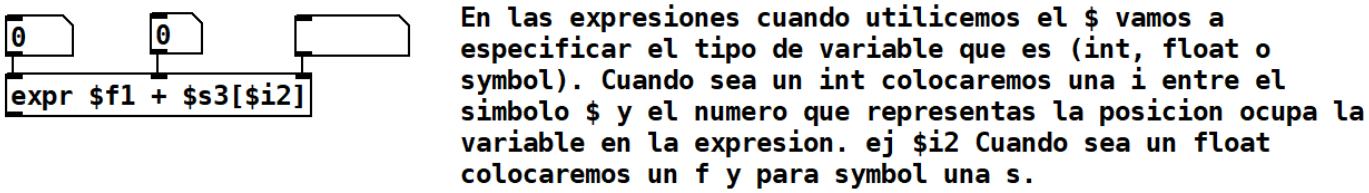


Figura 4. patch *Expr-explicacion.pd*

[**expr~**] es el objeto expr que nos va a permitir tratar con **señales**, tiene una variable input específica para señales '**$v#**'. En este caso colocamos una **v** entre el símbolo $ y el valor que indica la posición. Abre el patch ***Expr-explicacion.pd*** para probar el objeto "expr"

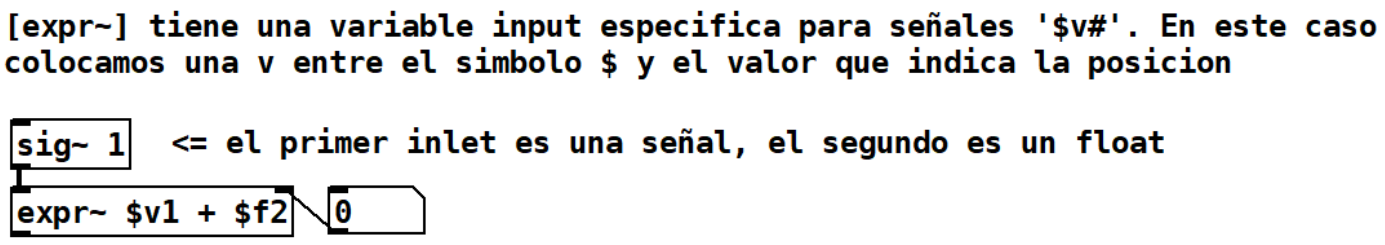


Figura 5. patch *Expr-explicacion.pd*

Como sabemos la onda de sierra es una rampa lineal que va de 0 a 1 en un tiempo determinado. Esta rampa pasa por valores intermedios entre 0 y 1, a la mitad de la rampa en concreto se encontrará en 0.5. La **onda rectangular** que queremos crear ahora no pasa por ningún valor intermedio, va **de 0 a 1 inmediatamente**. Para convertir esta rampa lineal en un salto vamos a decirle al objeto expr~ que la condición se cumple cuando los valores son superiores a 0.5. por lo tanto, todos los valores de 0 a 0.5 pasaran a ser un 0 y que todos los valores de 0.5 a 1 pasaran a ser un 1, ya que, verdadero es igual a 1 y falso es igual a 0:

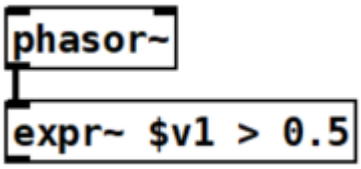


Figura 6. El objeto "phasor~" envía una señal que sera filtrada por el objeto "expr~ $v1 > 0.5"

$v1 sera la señal que el "phasor~" en via al inlet de "expr~". Cuando los valores generados por el phasor sean mayores que 0.5 la expresión devolverá por su output un 1, cuando los valores sean menores que 0.5 devolverá por su output un 0.

Copyright 2026

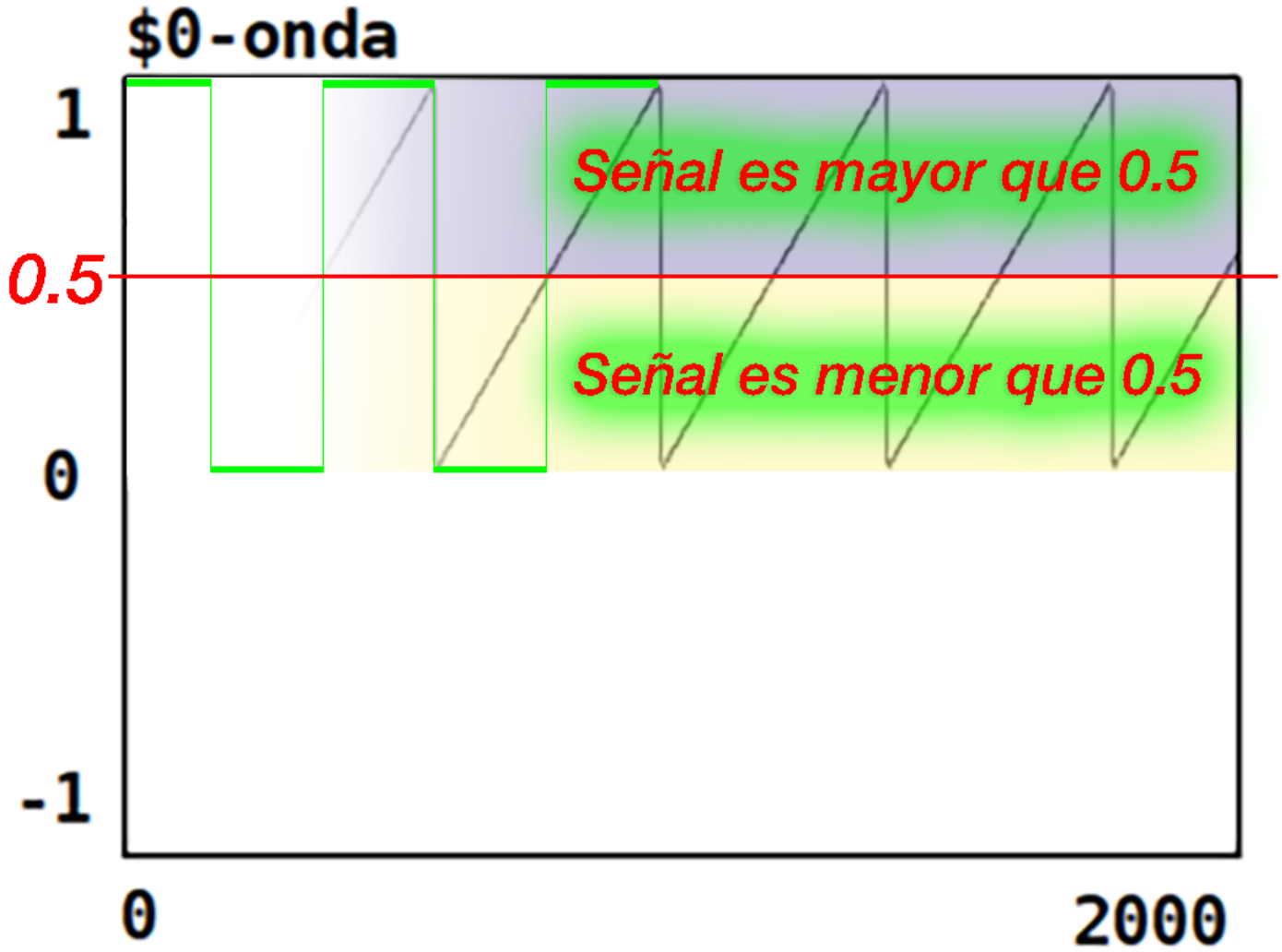


Figura 7. Representación de como el objeto "expr~ $v1 > 0.5" transforma una onda de sierra en una onda cuadrada.

¿Os acordáis de la metáfora del rio Ebro que utilizábamos en la lección Programación en genera... | Librería CATEDU?

El objeto "expr" y el operador ">" realizarían la acción de evaluar la altura del agua y cuando desciende por debajo de 50 abrirían la compuerta del embalse.

Ahora tenemos una **onda rectangular** que va **de 0 a 1**, luego su **amplitud** sera de **0.5**, por lo que vamos a **normalizarla** como hicimos anteriormente con la onda de sierra. La **multiplicaremos por 2** para obtener una amplitud de 1 y para colocar su centro en 0 le **restaremos 1**.

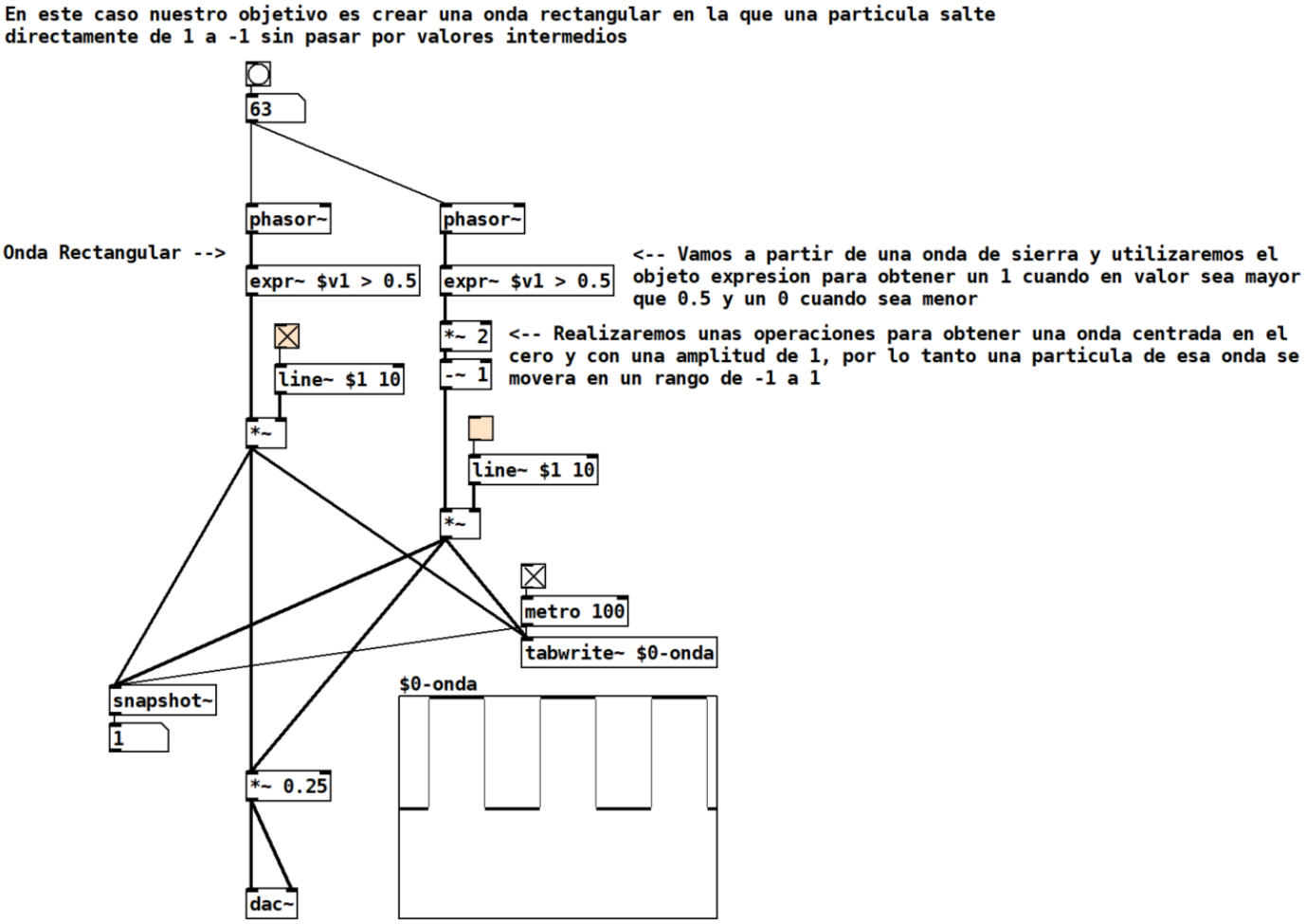


Figura 8. patch *Onda-Rectangular.pd*

Onda triangular

Su forma se compone de **rampas ascendentes y descendentes** que van **de -1 a 1 y de 1 a -1**. Para construirla vamos a partir de nuevo de una onda de sierra. A mitad de la progresión de esta onda de sierra vamos a **invertir la pendiente** de la progresión, de manera que al llegar a **0.5** en lugar de seguir subiendo hasta 1, **comience a bajar** hasta 0 con la misma pendiente pero invertida. Para ello vamos a utilizar dentro del objeto **expr~** el condicional **if**, que nos va a permitir configurar los outputs en función de si la condición se cumple o no. su estructura es la siguiente: **if(condicion, verdadero, falso)**. Nuestra condición va a se que la señal sea mayor que 0.5, cuando la condición se cumpla la expresión va a devolver la inversa de la señal (1-señal) y cuando la condición no se cumpla devolverá la señal que ha entrado sin ninguna modificación. La expresión sera la siguiente:

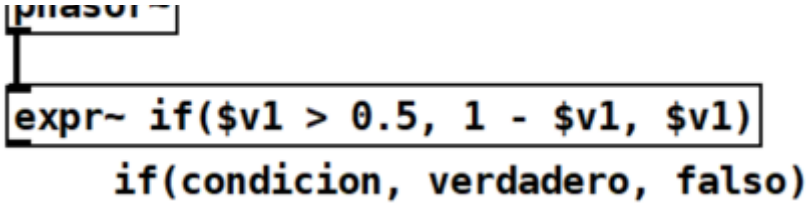


Figura 9. El objeto "phasor~" envía una señal que sera filtrada por el objeto "expr~ if($v1 > 0.5, 1 - $v1, $v1)".

$v1 sera la señal que el "phasor~" en via al inlet de "expr~". Cuando los valores generados por el phasor sean mayores que 0.5 la expresión devolverá la inversa de la señal (1-señal), cuando los valores sean menores que 0.5 devolverá la señal.

"expr~ if($v1 > 0.5, 1 - $v1, $v1)"

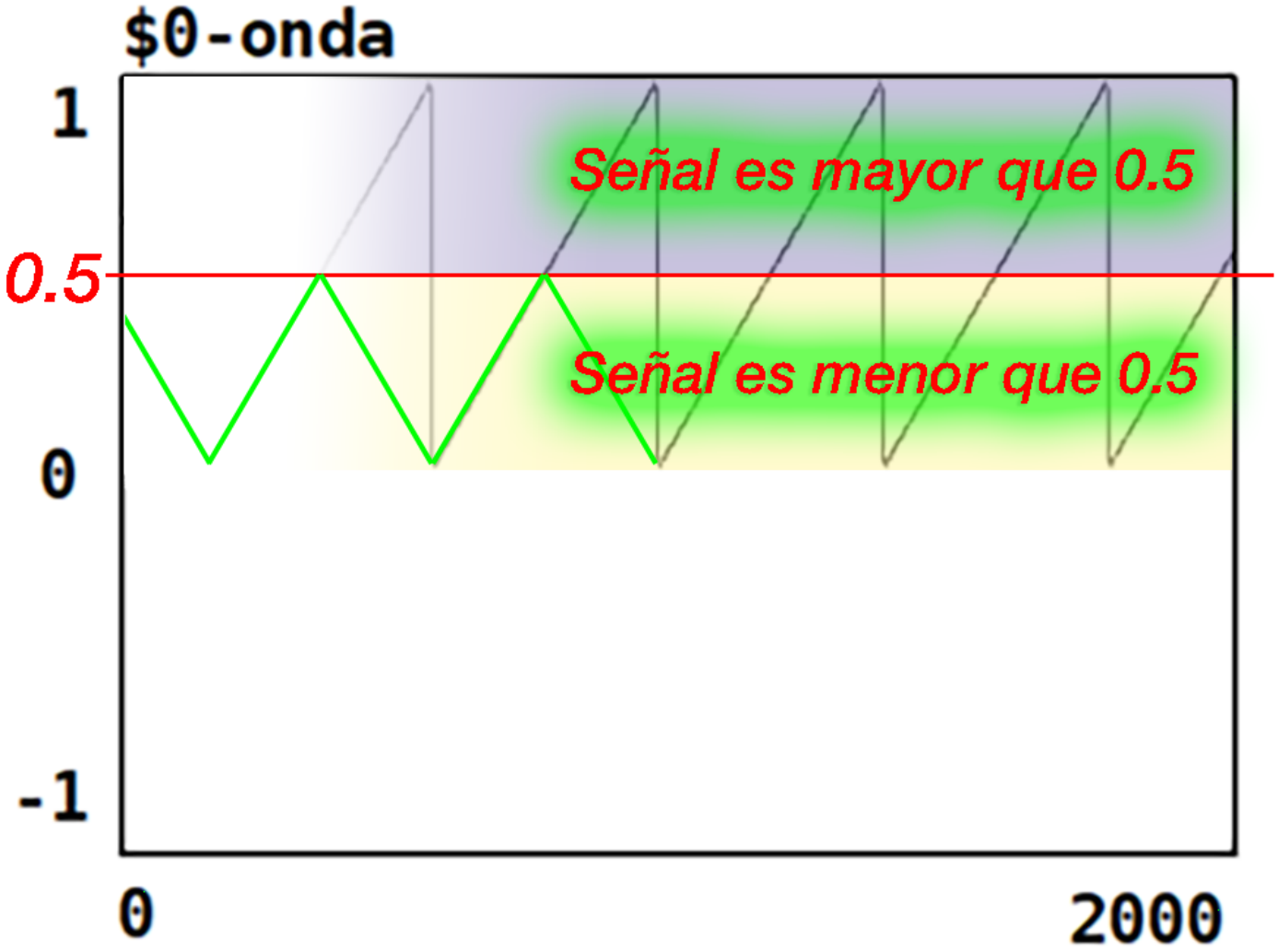


Figura 10. Representación de como el objeto "expr~ if($v1 > 0.5, 1 - $v1, $v1)" transforma una onda de sierra en una onda cuadrada.

Ahora tenemos una **onda triangular** que va **de 0 a 0.5**, luego su **amplitud** sera de **0.25**, por lo que vamos a **normalizarla** como hicimos anteriormente con la onda de sierra y la onda rectangular. La **multiplicaremos por 4** para obtener una amplitud de 1 y para colocar su centro en 0 le **restaremos 1**.

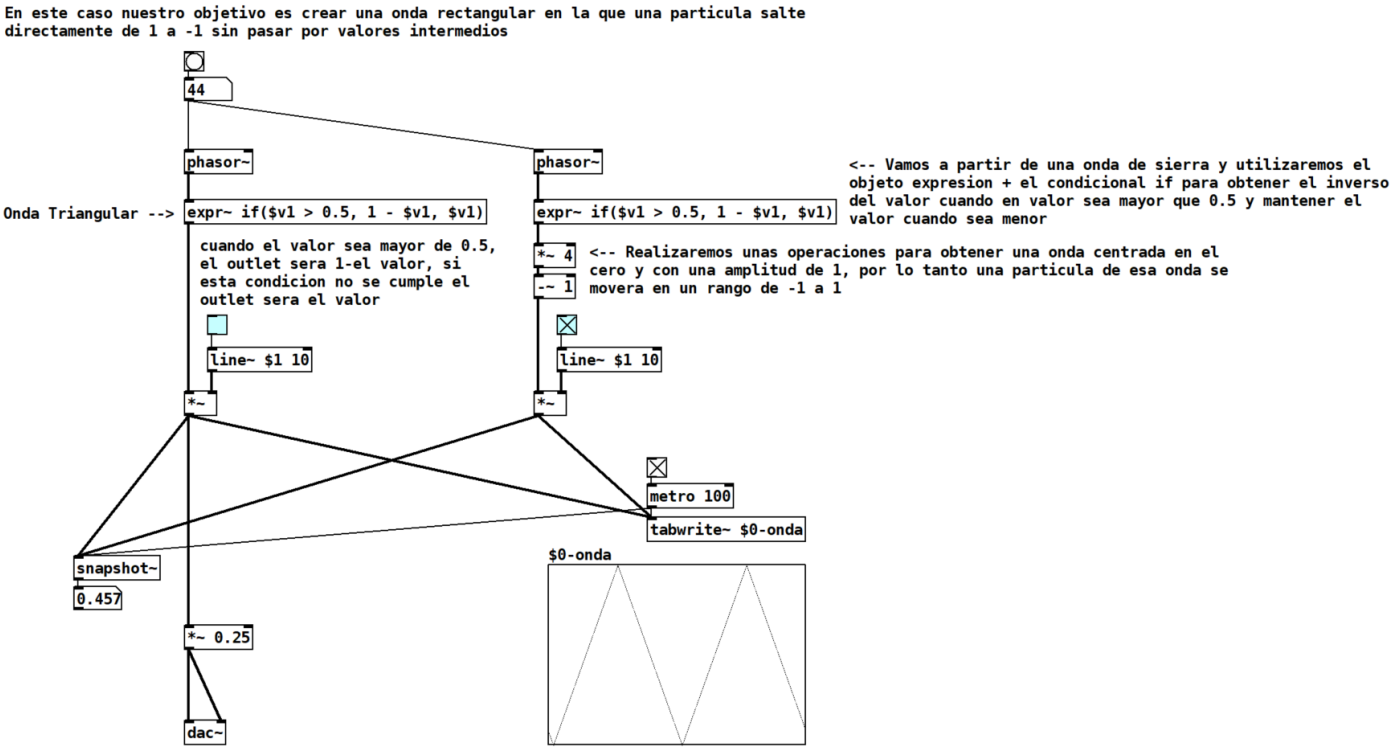


Figura 11. patch *Onda-Triangular.pd*

¿Alguna duda de las ondas que acabamos de ver? ¿Preparados para oscilar?

https://giphy.com/embed/bTJcxLULJMeZi

Figura 12. Perro oscilando.

En la **carpeta de material** de esta lección tenéis todos los patches que hemos visto y también un patch con todas las ondas (***Osciladores.pd***) para que comparéis y experimentéis un poco con ellas.

Figuras:

https://giphy.com/clips/storyful-lady-gaga-poker-face-coach-beard-pS8uMLP7sgXmK8d3Mr

Figura 2. patch *Onda-Sinusoidal.pd*

Figura 3. patch *Onda-Sierra.pd*

Figura 4. patch *Expr-explicacion.pd*

Figura 5. patch *Expr-explicacion.pd*

Figura 6. El objeto "phasor~" envía una señal que sera filtrada por el objeto "expr~ $v1 > 0.5"

Figura 7. Representación de como el objeto "expr~ $v1 > 0.5" transforma una onda de sierra en una onda cuadrada.

Figura 8. patch *Onda-Rectangular.pd*

Figura 9. El objeto "phasor~" envía una señal que sera filtrada por el objeto "expr~ if($v1 > 0.5, 1 - $v1, $v1)".

Figura 10. Representación de como el objeto "expr~ if($v1 > 0.5, 1 - $v1, $v1)" transforma una onda de sierra en una onda cuadrada.

Figura 11. patch *Onda-Triangular.pd*

Revision #1
Created 2023-02-17 11:30:16 CET by Julia del Río
Updated 2023-02-17 11:30:16 CET by Julia del Río

Copyright 2026